```
%This is embodiment #1
%Programmer: Carlos E. Davila
%programmer: Carlos E. Davila
%Dept. of Electrical Engineering, Southern Methodist University
%date of last modification: 12/7/99
clear;
randn('state',0);
lam1 = 0.8;
lam0 = 0.8;
N = 16;
r = 4;
T_max = N*2000;
  x(1:T_max/2) = cos(0.35*pi*[1:T_max/2]) + ...
      cos(0.78*pi*[1:T_max/2] + 0.35*pi);
   x(T_max/2+1:T_max) = cos(0.6*pi*[1:T_max/2]) + ...
      cos(0.8*pi*[1:T_max/2] + 0.35*pi);

x1_rec = [];
x_hat_rec = [];
ntr = 1;
b = 8;
for itr = 1:ntr,

itr
sig_n = 0.0001;

x_max = max(x);
x_min = min(x);
del = (x_max - x_min)/b;
Ro = eye(N)*0.001;
U=randn(N,4);
U=orth(U);
q_3 = U(:,4);
q_2 = U(:,3);
q_1 = U(:,2);
q_0 = U(:,1);
x_0 = x(N*(itr-1)+1:N*itr)';
x_0 = x_0 - mean(x_0);

Q_hat = real([q_3 q_2 q_1 q_0]);

R_hat = eye(N)*0.001;

%for it = 2:50,
for it = 1:2000,
if rem(it,100) == 0
      [it err(it-1)]
end

x_0 = x(N*(it-1)+1:N*it)';
%x_0 = x_0 - mean(x_0);
   v_0 = randn(N,1);

      Ro = lam0*Ro + x_0*x_0';
      [Vo Do] = eig(Ro);
      Do = diag(Do);
      [Do Io] = sort(Do);
      q_0o = Vo(:,Io(N));
      q_1o = Vo(:,Io(N-1));
      q_2o = Vo(:,Io(N-2));
      q_3o = Vo(:,Io(N-3));
```

```
        Qo_hat = [q_0o q_1o q_2o q_3o];

%Q1 = [Q_hat v_0];
%xh = floor(x_0/del)*del;
%xh = v_0 - Q_hat*Q_hat'*v_0;
%xh = xh/norm(xh);
%Q1 = [Q_hat sign(x_0)];

        Q1 = [Q_hat v_0];
        A = lam1*Q1'*R_hat*Q1 + Q1'*x_0*x_0'*Q1;
        B = Q1'*Q1;
        [V D] = eig(A,B);
        %[V D] = eig(A);
        D = diag(D);

        for n = 1:5
           if abs(imag(D(n))) > 0.001
              D(n) = 0;
           end
        end
        [Ds I] = sort(D);
%        alpha = [V(:,I(6)) V(:,I(5)) V(:,I(4)) V(:,I(3))];
        alpha = real([V(:,I(5)) V(:,I(4)) V(:,I(3)) V(:,I(2))]);
        Beta = Q1*alpha;
        %Beta = Q*V;
        q_0 = Beta(:,1);
        q_1 = Beta(:,2);
        q_2 = Beta(:,3);
        q_3 = Beta(:,4);
        x_hat = Q_hat*Q_hat'*x_0;
        Q_hat = [q_0/norm(q_0) q_1/norm(q_1) q_2/norm(q_2) q_3/norm(q_3)];

 Q_hat = Q_hat + flipud(Q_hat);
 Q_hat = orth(Q_hat);

        %R_hat = Q_hat*diag(flipud(Ds(2:5)))*Q_hat' + eye(N)*Ds(1);
        R_hat = Q_hat*diag(flipud(Ds(2:5)))*Q_hat' + eye(N)*Ds(1);
        P_Q = Q_hat*inv(Q_hat'*Q_hat)*Q_hat';

              P_Qo = Qo_hat*inv(Qo_hat'*Qo_hat)*Qo_hat';
        errq(it) =  norm(P_Q - P_Qo,'fro');
        x_hat_rec = [x_hat_rec x_hat'];
        err(it) = norm(x_0 - x_hat)^2;
 %   end
 end
 end%itr
```

A-2

```
%This is embodiment #2
%programmer: Carlos E. Davila
%Dept. of Electrical Engineering, Southern Methodist University
%date of last modification: 12/9/99
clear;
lam1 = 0.7;
lam0 = 0.7;
N = 32;
T_max = N*2000;
  x(1:T_max/2) = cos(0.3*pi*[1:T_max/2]) + ...
      cos(0.7*pi*[1:T_max/2] + 0.35*pi);
   x(T_max/2+1:T_max) = cos(0.6*pi*[1:T_max/2]) + ...
      cos(0.8*pi*[1:T_max/2] + 0.35*pi);

x1_rec = [];
x_hat_rec = [];
ntr = 1;
b = 8;
for itr = 1:ntr,

itr
sig_n = 0.0001;

x_max = max(x);
x_min = min(x);
del = (x_max - x_min)/b;
Ro = eye(N)*0.001;
U=randn(N,4);
U=orth(U);
q_3 = U(:,4);
q_2 = U(:,3);
q_1 = U(:,2);
q_0 = U(:,1);
x_0 = x(N*(itr-1)+1:N*itr)';
x_0 = x_0 - mean(x_0);

Q_hat = real([q_3 q_2 q_1 q_0]);
R_hat = eye(N)*0.001;
v_0 = randn(N,1000);

%for it = 2:50,
for it = 1:2000,
if rem(it,100) == 0
      [it err(it-1)]
end

x_0 = x(N*(it-1)+1:N*it)';
%x_0 = x_0 - mean(x_0);


      Ro = lam0*Ro + x_0*x_0';
      [Vo Do] = eig(Ro);
      Do = diag(Do);
      [Do Io] = sort(Do);
      q_0o = Vo(:,Io(N));
      q_1o = Vo(:,Io(N-1));
      q_2o = Vo(:,Io(N-2));
      q_3o = Vo(:,Io(N-3));
      Qo_hat = [q_0o q_1o q_2o q_3o];

      pow_max = 0;
```

```
            for m = 1:1000
      xh = v_0(:,m);

      Q1 = [Q_hat xh];

      A = lam1*Q1'*R_hat*Q1 + Q1'*x_0*x_0'*Q1;
      B = Q1'*Q1;
      [Vr Dr] = eig(A,B);
      Dr = diag(Dr);

      for n = 1:5
         if abs(imag(Dr(n))) > 0.001
            Dr(n) = 0;
         end
      end
      [Drs Ir] = sort(Dr);
      pow = sum(Drs(2:5));
      if pow > pow_max
         pow_max = pow;
         V = Vr;
         I = Ir;
         Ds = Drs;
         v_n = xh;
      end
      end%m

%      alpha = [V(:,I(6)) V(:,I(5)) V(:,I(4)) V(:,I(3))];
            alpha = [V(:,I(5)) V(:,I(4)) V(:,I(3)) V(:,I(2))];
            Q1 = [Q_hat v_n];

      Beta = Q1*alpha;
      %Beta = Q*V;
      q_0 = Beta(:,1);
      q_1 = Beta(:,2);
      q_2 = Beta(:,3);
      q_3 = Beta(:,4);
      Q_hat = [q_0/norm(q_0) q_1/norm(q_1) q_2/norm(q_2) q_3/norm(q_3)];
      x_hat = Q_hat*Q_hat'*x_0;
      R_hat = Q_hat*diag(flipud(Ds(2:5)))*Q_hat' + eye(N)*Ds(1);
      P_Q = Q_hat*inv(Q_hat'*Q_hat)*Q_hat';

            P_Qo = Qo_hat*inv(Qo_hat'*Qo_hat)*Qo_hat';
      %err(it) =  norm(P_Q - P_Qo,'fro')/ntr;
      x_hat_rec = [x_hat_rec x_hat'];
      err(it) = norm(x_0 - x_hat)^2;
      [it err(it)]
%   end
end
end%itr
```

```
%This is embodiment #4
%Programmer: Carlos E. Davila
%programmer: Carlos E. Davila
%Dept. of Electrical Engineering, Southern Methodist University
%date of last modification: 11/10/00
clear;
randn('state',0);
lam1 = 0.995;
sample_rate = 8000;
mse_max = 0.9e-2;
%mse_max = 5e-2;
N = 64;
M = 1024;
r = N;
T_max = N*2000;
%load x
load s18
%load err_s18;
%load m9
%x = err_s18';

x = x'/norm(x)*33.5326;


%wc = 0.6;
%h = [sin(wc*pi*[-16:16])./([-16:16]*pi)];
%h(17) = wc;
%x = filter(h,1,x);

bitrate = 0;
eval_min = 25e-3;
b_min = 3;
max_repeats = 2;
rep_rate = 6;
rate_0 = 4;
k_max = floor(N/2)+1;
k_max = N;
nstd = 6;
b = rate_0*ones(1,k_max);
x_fs_0 = 4*ones(1,k_max);

Do = 10*ones(1,N);
%  x(1:T_max/2) = cos(0.35*pi*[1:T_max/2]) + ...
%       cos(0.78*pi*[1:T_max/2] + 0.35*pi);
%   x(T_max/2+1:T_max) = cos(0.6*pi*[1:T_max/2]) + ...
%       cos(0.8*pi*[1:T_max/2] + 0.35*pi);
x1_rec = [];
x_hat_rec = [];
err_cnt = 0;
x_hat = zeros(N,1);


Q_hat=randn(N,r);
Q_hat=orth(Q_hat);

Lambda = diag([r:-1:1]/r);

R_hat = Q_hat*Lambda*Q_hat'*0.000000001;
v_0 = randn(N,M);

for m = 1:M,
    v_0(:,m) = v_0(:,m)/norm(v_0(:,m));
end

m_opt = zeros(1,length(x)/N);

for it = 1:length(x)/N,

      x_0 = x(N*(it-1)+1:N*it)';


      %Transmitter search for best search direction

   %Determine how many KLT coefficients to use
   %Q_hat=orth(Q_hat);

   mse = 100;
   y1 = Q_hat'*x_0;



     %update quantization parameters

    rate = rate_0;
    for k = 1:k_max,
       b(k) = max(rate +
 0.5*log2(Do(k)/max(prod(Do(1:k_max))^(1/k_max),eval_min)),0);
       b(k) = floor(b(k));
       b(k) = max(b(k),b_min);
       x_fs(k) = nstd*sqrt(Do(k));
    end

    y1 = quant2(y1,b,x_fs,k_max);

    x_hat_old = x_hat;

    k = 1;
       repeats = 0;
  while mse > mse_max & repeats < max_repeats%(used for N = 16)
  %  while mse > 1E-5
             x_hat = Q_hat(:,1:k)*y1(1:k);
             mse = norm(x_hat - x_0)^2/norm(x_0)^2;
          k = k + 1;
          if k == k_max+1 & mse > mse_max
             %     for m = 2:N,
         %           q = Q_hat(:,m);
         %           q = q - Q_hat(:,1:m-1)*Q_hat(:,1:m-1)'*q;
         %           q = q/norm(q);
         %         Q_hat(:,m) = q;
         %
             %     end%m
                        'missed'
            % k = k_max+1;
           %  y1 = Q_hat'*x_0;
             %     y1 = quant2(y1,b,x_fs,k_max);
          %mse = 0;

          repeats = repeats + 1;
          x_fs = x_fs_0;
             for k = 1:50,
             b(k) = (rep_rate-k/50*rep_rate/2);
             %b(k) = rep_rate;
          end
          b = floor(b);
         % b(1:10) = 6;
         % b(11:20) = 4;
         % b(20:64) = 1;
          %B = B + 1;
          y0 = Q_hat'*x_0;
          y1 = quant2(y0,b,x_fs,k_max);
          %norm(y0 - y1)
          if repeats < max_repeats
             k = 1;
          end

        end

     end


  r_opt = k-1; %this is the model order
  m_opt(it) = r_opt;
  for k = r_opt:-1:1,
     if norm(y1(k:r_opt)) == 0
        m_opt(it) = k-1;
     end
  end
     y1(r_opt+1:N) = zeros(N-r_opt,1);
  mo(it) = r_opt;

  err= x_hat - x_0;

  err_max = 100;
  for m = 1:M,
     v = v_0(:,m);
     alpha = v'*err;
     err_v = err - alpha*v;
     if norm(err_v) < err_max
        mvq_opt = m;
        err_hat = alpha*v;
        err_max = norm(err_v);
   %       figure(4)
   %       hold off
   %       plot(err)
   %       hold on
   %       plot(err_hat,'r')
   %       pause(0.001)
      end
   end
 %  x_hat = x_hat + err_hat;
x_frame = [x_hat_old' x_hat']';
for n = 1:N
   x_n  = x_frame(n+1:n+N);
   R_hat = lam1*R_hat + x_n*x_n';
end

%R_hat = lam1*R_hat + x_hat*x_hat';

            [Vr Dr] = eig(R_hat);
            Dr = diag(Dr);
        for n = 1:5
           if abs(imag(Dr(n))) > 0.001
             Dr(n) = 0;
           end
        end
              [Drs Ir] = sort(Dr);
              pow = sum(Drs(2:N));
                    V = real(Vr);
                    I = Ir;
                    Ds = Drs;

        for k = 1:r,
           Q_hat(:,k) = V(:,I(N-k+1))/norm(V(:,I(N-k+1)));
        end%k


     %   R_hat = Q_hat*diag(flipud(Ds))*Q_hat';


     %Receiver Processing


        %P_Q =
  Q_hat(:,1:r_opt)*inv(Q_hat(:,1:r_opt)'*Q_hat(:,1:r_opt))*Q_hat(:,1:r_opt)';

  %     P_Qo =
  Qo_hat(:,1:r_opt)*inv(Qo_hat(:,1:r_opt)'*Qo_hat(:,1:r_opt))*Qo_hat(:,1:r_opt)
  ';
   %  errq(it) =  norm(P_Q - P_Qo,'fro');
   mse_opt(it) = mse;

     if rem(it,1) == 0
        [it r_opt m_opt(it) repeats]
        figure(1)
              hold off
              plot(x_0)
              hold on
          plot(x_hat,'r')
          figure(2)
          hold off
          Do=flipud(Ds)*(1-lam1);
         % [Do_min k_max] = min(abs(Do-0.001));
          plot(real(Do(1:r_opt)))
          hold on
          plot(y1(1:r_opt).*y1(1:r_opt),'r')
          figure(3)
          plot(b)
              pause(0.001);
       end

       x_hat_rec = [x_hat_rec x_hat'];
      bitrate(it) = (sum(b(1:m_opt(it)))+1+log2(N))/(N/sample_rate);

           'bitrate:'
       [mean(bitrate) bitrate(it)]

      end%it
```

```
soundsc(x_hat_rec,sample_rate)

mse_tot = norm(x_hat_rec-
x(1:length(x_hat_rec)))^2/norm(x(1:length(x_hat_rec)))^2;
```